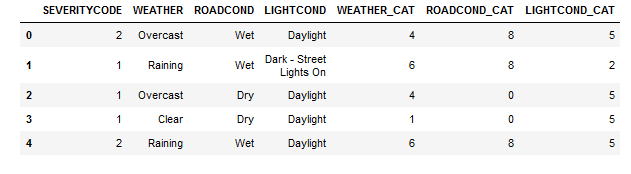
Extract Dataset & Convert

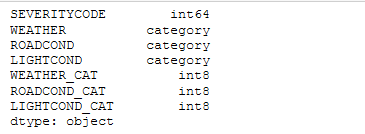
In it's original form, this data is not fit for analysis. For one, there are many columns that we will not use for this model. Also, most of the features are of type object, when they should be numerical type.

We must use label encoding to covert the features to our desired data type.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--chHdQUzJ--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/icadxir8nc4agrh2mv49.png)

With the new columns, we can now use this data in our analysis and ML models!

Now let's check the data types of the new columns in our dataframe. Moving forward, we will only use the new columns for our analysis.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--Z-rqpa6U--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/ho4kty0dz8gyqkvx170w.png)

**Balancing the Dataset**

Our target variable SEVERITYCODE is only 42% balanced. In fact, severitycode in class 1 is nearly three times the size of class 2.

We can fix this by downsampling the majority class.

[Alt Text](https://res.cloudinary.com/practicaldev/image/fetch/s--ab3RblIA--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/owt8a0ptc68nzkntehue.png)

Perfectly balanced.

Methodology

Our data is now ready to be fed into machine learning models.

We will use the following models:

**K-Nearest Neighbor (KNN)**

KNN will help us predict the severity code of an outcome by finding the most similar to data point within k distance.

**Decision Tree**

A decision tree model gives us a layout of all possible outcomes so we can fully analyze the concequences of a decision. It context, the decision tree observes all possible outcomes of different weather conditions.

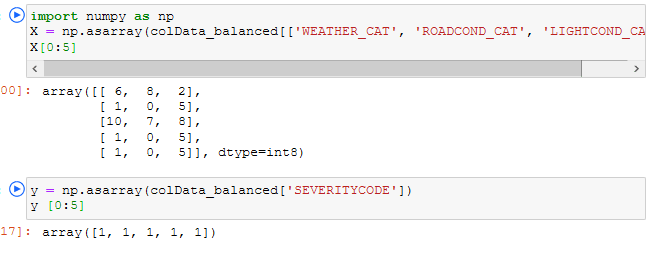
**Logistic Regression**

Because our dataset only provides us with two severity code outcomes, our model will only predict one of those two classes. This makes our data binary, which is perfect to use with logistic regression.

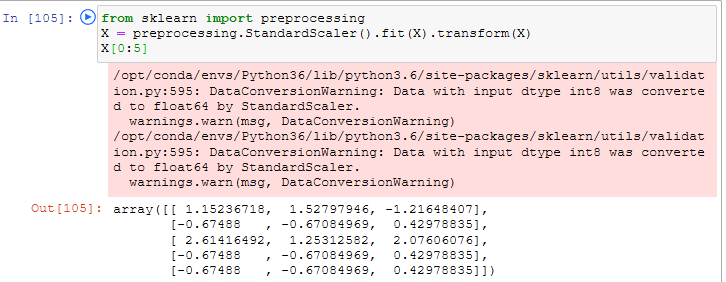
Let's get started!

Initialization

**Define X and y**

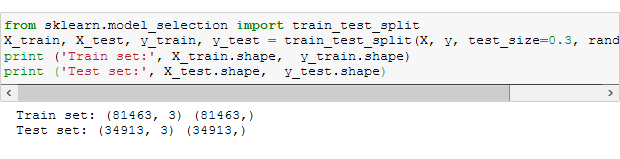
[](https://res.cloudinary.com/practicaldev/image/fetch/s---VOT9oo2--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/1a2o7uz8w0y6smfsgz1q.png)

**Normalize the dataset**

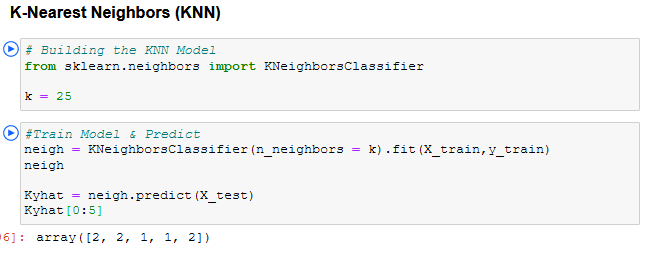
[](https://res.cloudinary.com/practicaldev/image/fetch/s--ZO0IcKw3--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/emcsck5jnf7o29dyqznl.png)

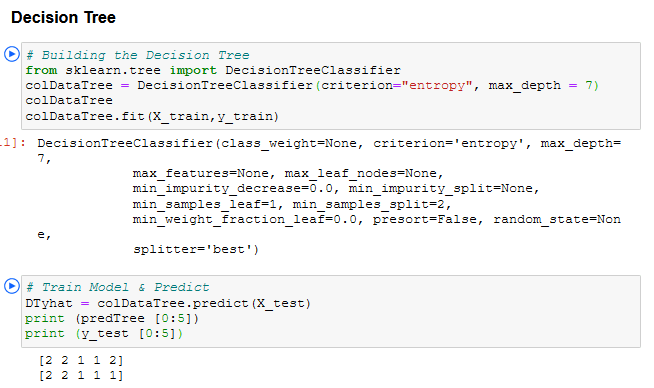
**Train/Test Split**

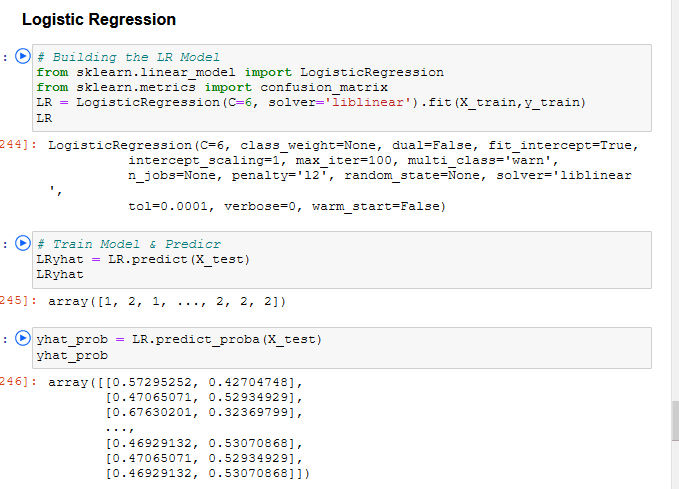
We will use 30% of our data for testing and 70% for training.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--e37BIsrW--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/j1218bmsdj4hwcbfajhj.png)

Here we will begin our modeling and predictions...

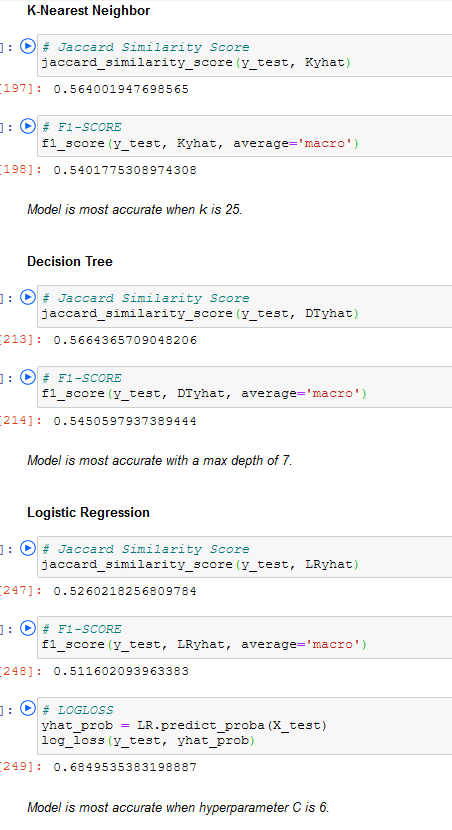
[](https://res.cloudinary.com/practicaldev/image/fetch/s--ZQWEYTJT--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/5hrdmvlbrfh98tzyn0am.png)

[](https://res.cloudinary.com/practicaldev/image/fetch/s--RbdUk-H_--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/vm860lwm2xlwxgmtx0og.png)

[](https://res.cloudinary.com/practicaldev/image/fetch/s---iRhfUHH--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/ksxnemhvsa6xteob1dpt.png)

Results & Evaluation

Now we will check the accuracy of our models.

[](https://res.cloudinary.com/practicaldev/image/fetch/s--lYC6RzUx--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/6cp9rajstv6hwewyfp0h.png)

Discussion

In the beginning of this notebook, we had categorical data that was of type 'object'. This is not a data type that we could have fed through an algoritim, so label encoding was used to created new classes that were of type int8; a numerical data type.

After solving that issue we were presented with another - imbalanced data. As mentioned earlier, class 1 was nearly three times larger than class 2. The solution to this was downsampling the majority class with sklearn's resample tool. We downsampled to match the minority class exactly with 58188 values each.

Once we analyzed and cleaned the data, it was then fed through three ML models; K-Nearest Neighbor, Decision Tree and Logistic Regression. Although the first two are ideal for this project, logistic regression made most sense because of its binary nature.

Evaluation metrics used to test the accuracy of our models were jaccard index, f-1 score and logloss for logistic regression. Choosing different k, max depth and hyparameter C values helped to improve our accuracy to be the best possible.

Conclusion

Based on historical data from weather conditions pointing to certain classes, we can conclude that particular weather conditions have a somewhat impact on whether or not travel could result in property damage (class 1) or injury (class 2).

Thank you for reading!